



# Motores MySQL

José Manuel Gómez

Departamento de Lenguajes y Sistemas Informáticos

[jmgomez@ua.es](mailto:jmgomez@ua.es)

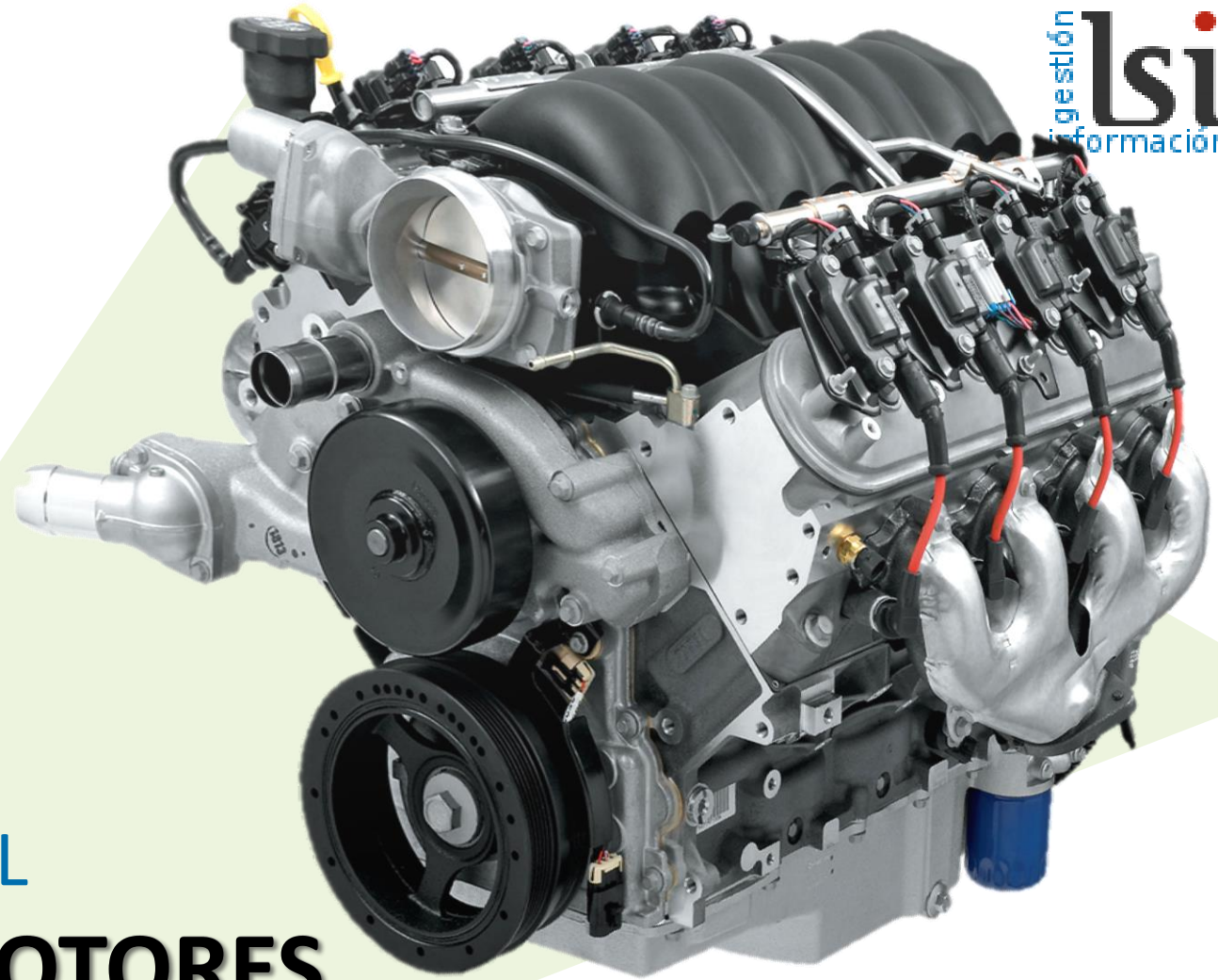


## Motores MySQL

# **INTRODUCCIÓN**

# INTRODUCCIÓN

- MySQL soporta distintos motores de tabla
- Cada tipo de motor tiene sus características
- Almacenan la información de forma distinta
- Recuperan la información de forma distinta
- Cada motor tiene su propia estructura de almacenamiento
- Se pueden combinar en una misma BD



## Motores MySQL

# **TIPOS DE MOTORES**



## TIPOS DE MYSQL

- InnoDB
- MyISAM
- MEMORY
- CSV
- ARCHIVE
- MERGE

# Motores MySQL

## **INNODB**

## INNODB

- Por defecto a partir de la versión 5.5
- Transacciones
- Reparación de errores
- Bloqueo a nivel de fila
- Mantiene la integridad de los datos
- Optimiza y ordena las tablas a partir de su CP
- Las búsquedas más eficientes por CP
- Diseñado para tratar con grandes volúmenes de datos
- Cambiar las columnas CP es un proceso muy costoso
  - Bloquea la tabla para lectura y escritura
- Almacena las CP en nodos B-Tree

## INNODB: ÍNDICES SECUNDARIOS

- Creación y borrado de índices secundarios eficientes
  - Incluso aconsejan hacerlo a posteriori
    - Excepto para claves ajenas
  - Bloquea la tabla para escritura pero no para lectura
- Un índice secundario tiene sólo el valor de la CP y la clave del índice secundario
  - Cuando cambias la CP también cambian los índices secundarios
- Por defecto también usa B-Trees.



# INNODB: COMPRESIÓN DE LA INFORMACIÓN

- InnoDB permite guardar tablas e índices con valores comprimidos
- Recomendado:
  - Columnas con textos y de lectura
- Es preciso especificar el almacenamiento separado:
  - SET GLOBAL innodb\_file\_per\_table=1;
  - SET GLOBAL innodb\_file\_format=Barracuda;
- Se define en el create table
  - CREATE TABLE t (*table\_definition*) ENGINE=InnoDB ROW\_FORMAT=COMPRESSED KEY\_BLOCK\_SIZE=8;
- O en el alter table
  - ALTER TABLE t ROW\_FORMAT=COMPRESSED KEY\_BLOCK\_SIZE=8;

# INNODB: AJUSTANDO LA COMPRESIÓN

- Ver qué tablas tengo comprimidas:
 

```
SELECT table_name, engine, row_format, table_rows
FROM information_schema.tables
WHERE table_schema = 'myproject';
```
- El valor de KEY\_BLOCK\_SIZE se obtiene empíricamente (1KB, 2KB, 4KB, 8KB, 16KB)
  - Determina el tamaño del buffer de lectura/escritura para comprimir/descomprimir los valores de las columnas
- Medir el tiempo de compresión/descompresión
  - `SELECT * FROM information_schema.innodb_cmp;`
  - `SELECT * FROM information_schema.innodb_cmp_reset;`
  - `SELECT * FROM information_schema.innodb_cmpmem;`
  - `SELECT * FROM information_schema.innodb_cmpmem_reset;`
- Recomendable para dispositivos SSD
- Deshaciendo compresión
  - `ALTER TABLE t ROW_FORMAT=COMPACT;`

## INNODB\_CMP Y INNODB\_CMPMEM

### ■ InnoDB\_CMPMEM y InnoDB\_CMPMEM\_RESET

- [https://dev.mysql.com/doc/refman/5.5/en/innodb-cmpmem-table.html#innodb-information-schema-innodb cmpmem columns table](https://dev.mysql.com/doc/refman/5.5/en/innodb-cmpmem-table.html#innodb-information-schema-innodb_cmpmem_columns_table)

### ■ INNODB\_CMP y INNODB\_CMP\_RESET

- [https://dev.mysql.com/doc/refman/5.5/en/innodb-cmp-table.html#innodb-information-schema-innodb cmp columns table](https://dev.mysql.com/doc/refman/5.5/en/innodb-cmp-table.html#innodb-information-schema-innodb_cmp_columns_table)

# Motores MySQL

## **MYISAM**

## MYISAM

- Almacenamiento y recuperación rápida
- No transacciones
- No integridad
- Dos modos:
  - Longitud estática
  - Longitud dinámica

## MYISAM: LONGITUD ESTÁTICA

### ■ Optimiza la velocidad

- Las columnas VARCHAR o VARBINARY ocupan lo mismo en todas las filas
- El valor de estas columnas está en la propia tabla
- Sólo funciona en tablas sin tipos TEXT y BLOB
- Optimización innecesaria (OPTIMIZE TABLE)
- Reparación muy sencilla y rápida (REPAIR TABLE)
- Valor por defecto
- CREATE TABLE t (*table\_definition*)  
ENGINE=MyIsam ROW\_FORMAT=FIXED;

## MYISAM: LONGITUD DINÁMICA

### ■ Optimiza el tamaño

- Las columnas CHAR, VARCHAR, BINARY o VARBINARY ocupan lo justo (si su tamaño es mayor que 4)
- El valor de estas columnas está aparte
- Reparación más complicada (REPAIR TABLE)
- Optimización necesaria en tablas que cambien mucho (OPTIMIZE TABLE)

■ CREATE TABLE t (*table\_definition*)  
ENGINE=MyIsam ROW\_FORMAT=DYNAMIC;

# Motores MySQL

## **MEMORY**



# MEMORY

- Tablas que se almacenan en memoria
- Acceso I/O hiperrápido
- Bloqueo a nivel de fila
- Todas las columnas de longitud fija
- No permite columnas TEXT o BLOB.
- Aconsejado para:
  - Pequeñas tablas
  - Velocidad crítica
  - Sólo lectura
- Es usual cargar los datos al principio
- ENGINE=memory

# Motores MySQL

## CSV

## CSV

- Crear archivos .csv con filas separadas por saltos de línea y columnas por un separador
  - "1","record one"
  - "2","record two"
- Este tipo de ficheros pueden ser importados por Excel o OpenOffice
- La modificación de la longitud de los valores de columnas de ancho variable puede ser muy costoso.
- ENGINE=csv

# Motores MySQL

## **ARCHIVE**

## ARCHIVE

- Para almacenar grandes cantidades de datos sin índices en un muy pequeño espacio
- Soporta sólo INSERT, UPDATE
- No soporta ORDER BY
- No soporta columnas BLOB
- Sólo permite índice en una columna AUTO\_INCREMENT
- El contenido es comprimido
- Las operaciones SELECT realizan un completo escaner de la tabla/s
- Aconsejado para:
  - archivos de registro
  - almacén de datos de lectura secuencial

# Motores MySQL

## **MERGE**

# MERGE

- Colección de tablas MyIsam idénticas
  - Mismas columnas e índices
- ```
CREATE TABLE t1 (
  a INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  message CHAR(20)) ENGINE=MyISAM;
CREATE TABLE t2 (
  a INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  message CHAR(20)) ENGINE=MyISAM;
INSERT INTO t1 (message) VALUES ('Testing'),('table'),('t1');
INSERT INTO t2 (message) VALUES ('Testing'),('table'),('t2');
CREATE TABLE total (
  a INT NOT NULL AUTO_INCREMENT,
  message CHAR(20), INDEX(a))
ENGINE=MERGE UNION=(t1,t2) INSERT_METHOD=LAST;
```

## MERGE: VENTAJAS E INCONVENIENTES

- Aumentar la velocidad
  - poniendo cada tabla en distintos discos
- Mejora la eficiencia de las búsquedas
  - eligiendo en qué tablas buscar
- Más fácil reparación
- Mapeado de distintas tablas inmediato
- Exceder el tamaño máximo de ficheros en el SO
- No se puede exceder el número máximo de filas
- Las tablas Merge no pueden tener índices únicos



## Gestión de Índices

# **PARTICIONAMIENTO**

# PARTICIONAMIENTO

- Dividir una tabla en distintos ficheros de acuerdo a unas reglas:
  - Por valor de una o varias columnas
  - Por el valor de una función
- Tipos de particionamiento
  - RANGE
  - LIST
  - COLUMNS
  - HASH
  - KEY
  - Subparticionamiento

## PARTICIONAMIENTO: RANGE

- Dividido por rangos de valores de una columna

```
CREATE TABLE articulo (
    id INT,
    descripcion VARCHAR(30),
    pvp DECIMAL(6,2)
) ENGINE=InnoDB
PARTITION BY RANGE(pvp) (
    PARTITION barato VALUES LESS THAN (1),
    PARTITION normal VALUES LESS THAN (100),
    PARTITION caro VALUES LESS THAN MAX_VALUE
);
```

## PARTICIONAMIENTO: LIST

- Dividido por una lista de valores de una columna

```
CREATE TABLE articulo (
    id INT,
    descripcion VARCHAR(30),
    tipo VARCHAR(20),
    pvp DECIMAL(6,2)
) ENGINE=InnoDB
PARTITION BY LIST(tipo) (
    PARTITION monitores VALUES IN ('tv', 'monitor', 'pantalla'),
    PARTITION musica VALUES IN ('radio', 'mp3', 'lector cd'),
    PARTITION informatica VALUES IN ('ordenador', 'portatil')
);
```

## PARTICIONAMIENTO: HASH

- Dividido por un Hash

```
CREATE TABLE articulo (
    id INT,
    descripcion VARCHAR(30),
    pvp DECIMAL(6,2)
) ENGINE=InnoDB
PARTITION BY HASH(id) PARTITIONS 4;
```

## PARTICIONAMIENTO: KEY

- Dividido por la clave primaria o índice único
- Utiliza MD5
- No permite índices NOT NULL

```
CREATE TABLE articulo (
    id INT PRIMARY KEY,
    descripcion VARCHAR(30),
    pvp DECIMAL(6,2)
) ENGINE=InnoDB
PARTITION BY KEY() PARTITIONS 4;
```

## PARTICIONAMIENTO: SUBPARTICIONAMIENTO

- <https://dev.mysql.com/doc/refman/5.5/en/partitioning-subpartitions.html>

## PARTICIONAMIENTO: VENTAJAS Y RESTRICCIONES

- Almacenar más datos de lo que permite el SO en un fichero
- Algunas consultas pueden ser más rápidas
  - Si alguna condición del WHERE coincide con la partición
- Se puede paralelizar algunas consultas
- Se puede distribuir en distintos discos
- Foreign keys no soportado
- PARTITION BY KEY() debe ser una columna entera



Crear una pequeña memoria la descripción de qué tipos de motores son recomendables para cada tabla de vuestra BD y por qué.